#1477

No. K-65/66

1

Technical Memorandum

MULVAR: A COMPUTER PROGRAM FOR FUNCTIONALIZATION IN SEVERAL VARIABLES

F. V. Reed
Computation and Analysis Laboratory

U. S. Naval Weapons Laboratory
Dahlgren, Virginia

16

U. S. NAVAL WEAPONS LABORATORY

TECHNICAL MEMORANDUM

November 1966

No. K-65/66

MULVAR: A COMPUTER PROGRAM FOR FUNCTIONALIZATION IN SEVERAL VARIABLES

by

F. V. Reed

Approved by:

Ralph A. Niemann

RALPH A. NIEMANN
Director, Computation and
Analysis Laboratory

While the contents of this memorandum are considered to be correct, they are subject to modification upon further study.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.

TABLE OF CONTENTS

NWL Technical Memorandum No. K-65/66

Initial Distribution:
K
K-1
KA
KG
KO
KP
KR
KW
KE (5)
KEM
KEMX (5)
MAL (6)
File

ABSTRACT

A general-purpose multivariable functionalization program is described in terms of its general capabilities and the information which it affords the user.

FOREWORD

The work reported herein was begun under WEPTASK RT8801001/210-1/54390001, and extended under WEPTASK RM3773001/210-1/WS470B02, in connection with the development of a general purpose functionalization program for use in ballistics.

INTRODUCTION

The problem of finding an analytical formula which satisfactorily reproduces experimental data or data which are the end-product of more or less complex computations is a recurring one; and the number of necessary or useful functional forms is boundless. The program described herein affords the user a wide variety of functional building blocks and a highly flexible means of assembling them into a structure which will serve his purpose. The available forms are listed below.

	f	$\frac{\partial f}{\partial a}$	$\frac{\partial f}{\partial b}$	$\frac{\partial f}{\partial c}$
1.	a	1		
2.	$a \cos bx_i$	$\cos bx_i$	$-ax_i \sin bx_i$	
3.	$a \sin bx_i$	$\sin bx_i$	$ax_i \cos bx_i$	
4.	$a\, e^{bx_i}$	e^{bx_i}	$ax_i\, e^{bx_i}$	
5.	$a\, x_i^b$	x^b	$ax_i^b \log x_i$	
6.	$a \cos^n bx_i$*	$\cos^n bx_i$	$-na\, x_i \cos^{n-1} bx_i \sin bx_i$	
7.	$a \sin^n bx_i$*	$\sin^n bx_i$	$na\, x_i \sin^{n-1} bx_i \cos bx_i$	
8.	$a \ln bx_i$	$\ln bx_i$	$\frac{a}{b}$	
9.	$a \cos bx_i \cos cx_j$	$\cos bx_i \cos cx_j$	$-ax_i \sin bx_i \cos cx_j$	$-ax_j \cos bx_i \sin cx_j$
10.	$a \cos bx_i \sin cx_j$	$\cos bx_i \sin cx_j$	$-ax_i \sin bx_i \sin cx_j$	$ax_j \cos bx_i \cos cx_j$
11.	$a\, e^{bx_i} \cos cx_j$	$e^{bx_i} \cos cx_j$	$ax_i\, e^{bx_i} \cos cx_j$	$-ax_j\, e^{bx_i} \sin cx_j$
12	$a\, x_i^b \cos cx_j$	$x_i^b \cos cx_j$	$a(\log x_i)\, x_i^b \cos cx_j$	$-ax_j\, (x_i^b) \sin cx_j$
13.	$a \sin bx_i \sin cx_j$	$\sin bx_i \sin cx_j$	$ax_i \cos bx_i \sin cx_j$	$ax_j \sin bx_i \cos cx_j$
14.	$a\, e^{bx_i} \sin cx_j$	$e^{bx_i} \sin cx_j$	$ax_i\, e^{bx_i} \sin cx_j$	$ax_j\, e^{bx_i} \cos cx_j$
15.	$a\, x_i^b \sin cx_j$	$x_i^b \sin cx_j$	$a\, (\log x_i)\, x_i^b \sin cx_j$	$ax_j\, (x_i^b) \cos cx_j$
16.	$a\, e^{bx_i}\, e^{cx_j}$	$e^{bx_i}\, e^{cx_j}$	$a\, x_i\, e^{bx_i}\, e^{cx_j}$	$ax_j\, e^{bx_i}\, e^{cx_j}$
17.	$a\, e^{bx_i}\, x_j^c$	$e^{bx_i}\, x_j^c$	$a\, x_i\, x_j^c\, e^{bx_i}$	$a\, (\log x_j)\, x_j^c\, e^{bx_i}$
18.	$a\, x_i^b\, x_j^c$	$x_i^b\, x_j^c$	$a\, x_i^b\, (\log x_i)\, x_j^c$	$a\, x_i^b\, (\log x_j)\, x_j^c$
19.	$a\, x_i\, x_j \cos bx_k$**			
20.	$a\, x_i\, x_j \sin bx_k$**			
21.	$a\, {x_i}^{bx_j}$	${x_i}^{bx_j}$	$a(x_j \ln x_i){x_i}^{bx_j}$	

* $n \geq 2$, an integer.

**Available for use in the linear and computational modes only.

The partial derivatives of the forms with respect to the various parameters of fit are also listed; they are required in the program and are available, with some restrictions, for general use. The letters, a, b, c, are generic, of course, and are input quantities to be used, determined or improved as the case may be. The letters x_i and x_j are input variables, dependent or independent, at the disposal of the user. The number of input variables, the number of terms in the fitting-function and the number of data points which can be used depend on the dimensioning of the program to fit the internal memory of the computer.

GENERAL PLAN

A simple way to grasp the general capability of the program is to think of it in terms of

1. A table of, say, ten columns of memory of which any or all can be filled with input variables which can, so far as the machinery of the program is concerned, be used as dependent or independent variables;

2. A catalog of functions and their derivatives with respect to the parameters of fit;

3. A computational "black box" which

a. Evaluates a selection of functions from the catalog and, when it is appropriate, their derivatives, at the values of selected variables in the table to form an array of computed values which will be called the φ-matrix;

b. Multiplies the φ-matrix by its transpose, thus forming normal equations; and

c. Solves the normal equations to give the parameters of the fit.

Whenever a fit is made the printed end-output lists the parameters which were found, the average of the absolute values of the residuals of the fit, the square of the sum of the residuals, the summed squares and variance of the residuals; the average of the absolute values of the relative error at each point, expressed as a percent, is also given with the numerically largest residual and the point at which it occurs.

The phrase "end-output" is used above because there are several optional print-outs; those just listed are automatically given whenever a fit, as distinguished from an auxiliary computation, is made.

The program uses the central machinery for producing the normal matrix to give two allied modes of operation for functionalization: a linear, or true least-squares, mode and a non-linear or "differential correction" mode which will herein be called the *iterative mode*. A natural by-product is the computational mode which is useful both as an auxiliary in producing a fit and producing tables of functions for plotting or other uses.

THE LINEAR MODE

The linear mode is the simplest least-squares type of operation and produces the "least-squares best" fit by determining the coefficients a_i in functions of the form

$$y = a_1\varphi_1 + a_2\varphi_2 + \dots a_n\varphi_n \qquad (1)$$

in which the φ_i can be drawn in any manner from the list already given. The arguments of the *φ's* may be any input variable. It will be noticed that the listed functions individually contain either one or two variables, e.g., type 2 is $\underline{a \cos bx_i}$ in which the subscript *i* implies that *x* can be any input variable; type 9 in the list is $\underline{a \cos bx_i \cos cx_j}$ in which $\underline{x_i}$ and $\underline{x_j}$ can be any input variables or the same variable (see type 6). The letters *a*, *b*, and *c* are arbitrary input quantities in any case, but in the linear mode the "best" *a's* are found by the program. The customer decides in which column of the data-table he wishes to place a particular variable and then calls for it by number.

A series of cosines of a single argument, for example, would be set up by repetitive use of type 2 in the function-list. Suppose that the values of the common argument of the trigonometric terms have been put into column five, then the cosine series

$$y = \cos x + \cos 2x + \cos 3x + \cos 7x \qquad (2)$$

would be called from the program by a card showing

2 5, 2 5, 2 5, 2 5

based on type 2, i.e., $\underline{a \cos bx_i}$, and variable number 5. The desired harmonics would be called by another card showing

1 1, 1 2, 1 3, 1 7

which give the requisite values of a and b in $\underline{a \cos b_x}$. Although the inputs *a*, *b*, *c* in the forms are arbitrary, some values must always be given.

Examination of the list of functions shows that single and double Fourier series and the redoubtable least-squares polynomial are immediately available. Certain options to be discussed later allow approximations in terms of various types of polynomials such as those of Hermite, Legendre, Chebychev, etc. The number of terms of such expansions depends primarily on the number of slots allowed for input variables. Various standard multivariable expansions can be concocted. The coding of the program has been laid out in such a way as to allow the list of basic functions to be altered on an ad hoc basis. For example, the general-purpose program was readily adapted to allow 175 terms of double Fourier type, any or all of which might be modified by a positive or negative power of a non-angular variable; this adaptation handles a maximum of 9000 data-points.

Examples of types of operation which can be followed will be given later.

THE ITERATIVE MODE

The iterative mode of the program is a succession of applications of the differential correction technique allowing the determination not only of the linear coefficient, $\underline{a}$, appearing in the list of forms, but the parameters $\underline{b}$ and $\underline{c}$ as well.

The technique used (Reference 1) calls for the first differential of the fitting-function with respect to the various parameters; and these are catalogued internally, so to speak, in the program: when the control calling for the iterative mode is used the program automatically supplies the derivatives which are required.

The quantities which are found by solving the normal equations are used as corrections and are applied to the input values of the parameters; a new set of normal equations can be formed using the altered values to find new increments, and so on. The process can be terminated after a preassigned tolerance is met or after a preassigned number of iterations if the tolerance is not met. The initial values of the parameters have to be reasonably good, however, or the function is likely to blow up. A few ways of getting the process started on a reasonably sound basis will be discussed in a later section.

THE COMPUTATIONAL MODE

The computational mode involves no normal equations or solutions thereof. It is intended to extend the number of functions which are available to the user on the one hand, or to generate function-values to be used for purposes other than fitting. Generally the computational mode allows one to list or punch on cards practically any set of numbers which could enter into the computation of a fit.

OPTIONS AND OUTPUT

It is helpful in discussing options and output to speak of input "columns" and the columns of the φ-matrix, i.e., the columns of the matrix of the equations of condition.

I. Operational Options

A. Linear least squares

B. Differential correction, iterative mode

C. "Sum φ"; this option tells the computer to add the elements of the rows of the φ-matrix row by row for future use, without completing the least-squares routine.

II. Print-Only Options

A. Long print; this gives

1. a. The average of the absolute values of the residuals of fit,

   b. The variance of the residuals of fit,

   c. The algebraic sum of the residuals,

   d. The sum of the squared residuals,

   e. The square of the sum of the residuals,

   f. The signed absolute-maximum residual, the point number and value of input function at which it occurs.

   g. The average percent error of fit; the mean of the absolute values of (residual/input value) x 100.

   (Note: This information is given whenever a fit is made.)

2. a. The values of the function being fitted,

   b. The values of the input variables used in the fit,

   c. The residuals.

3. The φ-matrix.

4. The matrices of the normal equations

   (Note: This information is given after each step in the iterative mode, with the "correction" to each parameter, as well as the corrected parameters.)

B. The normal print; this gives

1. Item A.1. above

2. Item A.2. above

C. The short print; this gives

1. Item A.1. above.

(The input controls are always printed.)

III. Print-Punch-Store Option

The following items may be listed, card-punched or stored as input variables for subsequent use:

A. The fitted function; i.e., the computed values of the fit;

B. The row-by-row sum of the elements of the φ-matrix;

C. Any column of the φ-matrix.

IV. Miscellaneous Options

A. Segmentation Option. This option allows the user to select any block of consecutive data points for fitting, e.g., points 1-25, or 31-268.

B. Point-Selection Option. This option allows every n^{th} point of the entire set of input data or segment thereof to be used in a fit; the residuals, if called for, are computed over the entire block.

C. Carry-Over Options. These options allow:

1. A particular functional form to be used over several segments of data without a full set of controls for each segment;

2. The parameters as determined from one fit to be used (in the same functional form) in the next operation.

EXAMPLES AND DISCUSSION

The examples of uses to which the program has been put will not be given in terms of input or output numbers, but in terms of the ways in which the problems were handled.

In one case a shotgun approach to the following problem was used. Given some 90 sets of 72 similar but different physical measurements all made at equal intervals of a single independent angular variable it was desired to investigate whether these sets could be fitted with a Fourier type of function of sines and cosines, allowing as many as four harmonics. The version of the program which was used allowed a maximum of 700 data points in each of ten allowable variable-columns (and, incidentally, a maximum of 30 terms in the linear mode).

Seventy-two of the sets were handled in one pass through the computer by loading each of 9 input columns with 9 complete sets of measured values, and column 10 with 9 identical, complete sets of values of the angular variable; using the segmentation option and a carry-over option, all of the 81 stored sets were fitted successively with first harmonic, first and second harmonics, etc. The resulting fits were card-punched for later use in a plotting routine.

A simple example using the iterative mode is furnished by the problem of functionalizing the speed of a projectile in terms of its distance from the muzzle of the gun. The function used was

$$v = ae^{-ks}$$

v being speed and s distance from muzzle. It was desired in this case to find suitable values of a and k. From the general knowledge available, a good value of k was at hand and a plausible value of v could be estimated from the data, so that the sometimes knotty problem of finding initial values for the iterative process was easily solved.

A problem which allows the program a little more scope is furnished by a particular approach to the functionalization of bombing-table ranges. The ranges to be reproduced are taken as functions of height, speed and angle of dive or climb of the aircraft at the time of release of the bomb. A function which serves nicely for one type of bomb involves finding values of a_1, a_2, ..., a_6 in

$$R = a_1 v \cos a_2 \theta \exp(a_3 + a_4 v + a_5 \frac{v^{\frac{1}{2}}}{h} + a_6 \frac{h^{\frac{1}{2}}}{v})$$

A step-by-step technique for finding the six parameters in this function will be given; the reasons for the operations are not relevant. Using data at $\theta = 0$ (level flight),

1. Compute R/V (type 18 in the list of functions) and store;

2. Compute ln R/V and store (type 8);

3. Fit $\ln R/V = a_3 + a_4 + a_5 \frac{v^{\frac{1}{2}}}{h} + a_6 \frac{h^{\frac{1}{2}}}{v}$ (types 1, 5, 18);

   Using data based on, say, dive angles, and the parameters found in Step 3;

4. Compute $\exp(a_3 + a_4 v + a_5 \frac{v^{\frac{1}{2}}}{h} + a_6 \frac{h^{\frac{1}{2}}}{v})$ and store;

5. Compute $R/[V \exp(a_3 + a_4 v + a_5 \frac{v^{\frac{1}{2}}}{h} + a_6 \frac{h^{\frac{1}{2}}}{v})]$ and store;

6. Fit the result of step 5 with $a_1 \cos a_2\theta$ (type 2) using the iterative mode to evaluate a_1 and a_2.

If the total number of data points does not exceed the capacity of the program, the process can be completed in a single pass through the computer.

It will be apparent that so far as the linear mode of operation is concerned, none of the products, types 6, 7 and 9-20, is logically necessary; but in order to keep the entire program with its many arrays within the internal memory of the computer, and at the same time amendable to FORTRAN coding, one has to start and stop somewhere. If ten columns of n locations are allowed for input or computed quantities and a maximum of 30 terms in the fit, 41n locations are gone without any coding. However, there will be available for storing composite functions ten columns less the number of input variables.

The picture is somewhat different in the case of the iterative mode. Each parameter of fit gives rise to a column of the φ-array by way of the differential. Hence 30 available columns in the φ-array permit the finding of thirty parameters which may be involved in ten three-parameter functionals, or fifteen two-parameter types, or however. The whole scheme could doubtless be greatly improved at the level of computer-system development.

Use of the iterative mode must be used with caution. The technique of differential correction calls for the evaluation of the derivatives, shown in the list of functions, at some set of values of the parameters at each step of the iteration as they are inexorably ground out by the computer; and the only control one has over the process is in the choice of the first set, which is an input, and this choice is accordingly very important. One-term fits which involve powers or exponentials of a variable or products thereof can be managed by selecting a data point for each parameter and setting up two or three linear equations by transforming to logarithms; these equations can then be solved for the parameters. Lacking any clue at all, a cut and try approach can be taken by selecting a data point for each parameter - or several such sets if there is a large amount of data covering wide ranges of values of the variables - and producing a set of values for trial or examination by calling for a single iteration based on input parameters set equal for unity.

A systemmatic analysis of the questions involved here does not seem to have been carried out; it would be a major enterprise, but would be a valuable advancement of the art.

REFERENCE

1. F. V. Reed, Notes on Some Applications of the Principle of Least Squares to the Functionalization of Empirical Data, NWL Tech Memo No. K-93/65.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
U. S. Naval Weapons Laboratory
Dahlgren, Virginia

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
MULVAR: A COMPUTER PROGRAM FOR FUNCTIONALIZATION IN SEVERAL VARIABLES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*
Reed, F. V.

6. REPORT DATE
November 1966

7a. TOTAL NO. OF PAGES
13

7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
TM K-65/66

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

A general-purpose multivariable functionalization program is described in terms of its general capabilities and the information which it affords the user.

UNCLASSIFIED
Security Classification